

VERIFICATION OF TRANSLATION

I, Ayumi MORITA

of JFC Bldg., 68-3 Kandaneibeicho, Chiyoda-ku, Tokyo, Japan

declare that I am well acquainted with both the Japanese and English languages, and that the attached is an accurate translation, to the best of my knowledge and ability, of the Japanese Patent Application No. 2002-305856.

Signature

Ayumi Morita

Ayumi MORITA

Date

2, 25, 2009

[Document Name] Request for Patent Application

[Reference Number] 02P00010

[Special Instruction] Patent Application Eligible for Application of the Provisions set forth in Patent Law Section 30(1)

[Filing Date] October 21, 2002

[Destination] Patent Office Commissioner

[IPC] A63F 13/12

[Inventor]

[Address] c/o Square Co., Ltd., Arco Tower, 1-8-1
Shimomeguro, Meguro-ku, Tokyo

[Name] Yasumi MATSUNO

[Inventor]

[Address] c/o Square Co., Ltd., Arco Tower, 1-8-1
Shimomeguro, Meguro-ku, Tokyo

[Name] Yuichi MURASAWA

[Inventor]

[Address] c/o Square Co., Ltd., Arco Tower, 1-8-1
Shimomeguro, Meguro-ku, Tokyo

[Name] Shutaro YOKOYAMA

[Applicant]

[ID Number] 391049002

[Name] Kabushiki Kaisha Square

[Representative]

[ID Number] 100104916

[Attorney]

[Name] Satoshi KOMIZO

[Fees]

[Deposit Account Number] 153306

[Fee Paid] 21,000yen

[List of Submitted Materials]

[Document Name]	Specification	1
-----------------	---------------	---

[Document Name]	Drawings	1
-----------------	----------	---

[Document Name]	Abstract	1
-----------------	----------	---

[Number of General Power of Attorney]	9806805
---------------------------------------	---------

[Need of Proof]	Yes
-----------------	-----

[Document Name] SPECIFICATION

[Title of the Invention]

GAME APPARATUS, METHOD FOR ADVANCING GAME, PROGRAM AND
RECORDING MEDIUM

[Claims]

[Claim 1]

A game apparatus that advances a game in accordance with a player's instruction, comprising:

progress degree detecting means for detecting a degree of a game progress;

rule determining means for determining a rule applicable when the player advances the game according to the degree of the game progress detected by the progress degree detecting means;

instruction input means for inputting a desired instruction of the player to advance the game;

rule violation determining means for determining whether a violation of the rule determined by the rule determining means occurs based on the instruction input by the player from the instruction input means; and

penalty processing means for imposing a predetermined penalty in the game progress when the rule violation determining means determines that the violation occurs.

[Claim 2]

The game apparatus according to claim 1, wherein the rule violation determining means comprises means that determines whether the instruction itself input by the player from the instruction input means is against the rule determined by the rule determining means.

[Claim 3]

The game apparatus according to claim 1 or 2, further comprising instruction

executing means for executing processing according to the instruction input by the player from the instruction input means,

wherein the rule violation determining means comprises means that determines whether a processing result of the instruction executing means based on the instruction input by the player from the instruction input means is against the rule determined by the rule determining means.

[Claim 4]

The game apparatus according to any one of claims 1 to 3, wherein the rule applicable when the game advances is divided into a plurality of groups, and

the rule determining means determines a rule that belongs to a group according to the degree of the game progress detected by the progress degree detecting means as the rule applicable when the game advances.

[Claim 5]

The game apparatus according to any one of claims 1 to 4, wherein a number of rules applicable when the game advances is two or more, and

the rule determining means increases the number of rules applicable when the game advances as the degree of the game progress detected by the progress degree detecting means advances.

[Claim 6]

The game apparatus according to any one of claims 1 to 5, wherein the instruction input means inputs an instruction to operate a player character,

the game apparatus further comprises characteristic value storage means for storing a characteristic value of the player character that varies according to circumstances of the game progress and exerts an influence on the game progress, and

the penalty processing means imposes the predetermined penalty by varying the characteristic value of the player character stored in the characteristic value storage means.

[Claim 7]

The game apparatus according to any one of claims 1 to 6, wherein the instruction input means inputs a type of operation performed by the player character, which is allowed to perform a plurality of types of operations, and

the penalty processing means imposes the predetermined penalty by limiting the types of operations which the player character is allowed to perform.

[Claim 8]

The game apparatus according to claim 6 or 7, wherein a number of player characters to which the operation is allowed to be instructed from the instruction input means is two or more,

the rule violation determining means comprises means that judges a player character that causes the violation of the rule determined by the rule determining means based on the instruction input from the instruction input means among the plurality of player characters, and

the penalty processing means imposes the predetermined penalty on the player character that causes the violation of the rule.

[Claim 9]

The game apparatus according to any one of claims 1 to 8, further comprising item storage means for storing items given to the player with the game progress,

wherein the penalty processing means imposes the predetermined penalty by deleting a predetermined item among the items stored in the item storage means.

[Claim 10]

The game apparatus according to claim 9, wherein the item storage means stores the items given to the player by classifying the items into groups determined based on a value in the game progress,

the rule violation determining means also determines a degree of the violation of the rule when the rule violation determining means determines that the violation of the

rule determined by the rule determining means occurs based on the instruction input from the instruction input means, and

the penalty processing means deletes an item which belongs to a group according to the degree of the violation of the rule among the items stored in the item storage means.

[Claim 11]

A game apparatus that advances a game in accordance with a player's instruction, comprising:

rule storage means for storing a rule applicable when the player advances the game;

instruction input means for inputting a desired instruction of the player to advance the game;

rule violation determining means for determining whether a violation of the rule stored in the rule storage means occurs based on the instruction input by the player from the instruction input means;

violation history storage means for storing a history of violations determined by the rule violation determining means; and

penalty processing means for imposing a predetermined penalty in a game progress with a degree according to the violation history stored in the violation history storage means when the rule violation determining means determines that the violation occurs.

[Claim 12]

The game apparatus according to claim 11, wherein the rule violation determining means comprises means that determines whether the instruction itself input by the player from the instruction input means is against the rule stored in the rule storage means.

[Claim 13]

The game apparatus according to claim 11 or 12, further comprising instruction executing means for executing processing according to the instruction input by the player from the instruction input means,

wherein the rule violation determining means comprises means that determines whether a processing result of the instruction executing means based on the instruction input by the player from the instruction input means is against the rule stored in the rule storage means.

[Claim 14]

A method for advancing a game executed by a computer apparatus, comprising the steps of:

detecting a degree of a game progress;

determining a rule applicable when a player advances the game according to the detected degree of the game progress;

inputting a desired instruction of the player to advance the game;

determining whether a violation of the determined rule occurs based on the instruction input by the player; and

imposing a predetermined penalty in the game progress when it is determined that the violation of the rule occurs.

[Claim 15]

A method for advancing a game executed by a computer apparatus that stores a rule applicable when a player advances the game, comprising the steps of:

inputting a desired instruction of the player to advance the game;

determining whether a violation of the stored rule occurs based on the instruction input by the player;

storing a history of the determined violations of the rule; and

imposing a predetermined penalty in a game progress with a degree according to the stored history of the violation of the rule when it is determined that the violation of

the rule occurs.

[Claim 16]

A program for advancing a game which program is executed by a computer apparatus, allowing the computer apparatus to execute the steps of:

detecting a degree of a game progress;

determining a rule applicable when a player advances the game according to the detected degree of the game progress;

inputting a desired instruction of the player to advance the game;

determining whether a violation of the determined rule occurs based on the instruction input by the player; and

imposing a predetermined penalty in the game progress when it is determined that the violation of the rule occurs.

[Claim 17]

A program for advancing a game which program is executed by a computer apparatus that stores a rule applicable when a player advances the game, the program allowing the computer apparatus to execute the steps of:

inputting a desired instruction of the player to advance the game;

determining whether a violation of the stored rule occurs based on the instruction input by the player;

storing a history of the determined violations of the rule; and

imposing a predetermined penalty in a game progress with a degree according to the stored history of the violation of the rule when it is determined that the violation of the rule occurs.

[Claim 18]

A computer-readable storage medium on which a program for advancing a game is recorded, which program is executed by a computer apparatus, the program allowing the computer apparatus to execute the steps of:

detecting a degree of a game progress;
determining a rule applicable when a player advances the game according to the detected degree of the game progress;
inputting a desired instruction of the player to advance the game;
determining whether a violation of the determined rule occurs based on the instruction input by the player; and
imposing a predetermined penalty in the game progress when it is determined that the violation of the rule occurs.

[Claim 19]

A computer-readable storage medium on which a program for advancing a game is recorded, which program is executed by a computer apparatus that stores a rule applicable when a player advances the game, the program allowing the computer apparatus to execute the steps of:

inputting a desired instruction of the player to advance the game;
determining whether a violation of the stored rule occurs based on the instruction input by the player;
storing a history of the determined violations of the rule; and
imposing a predetermined penalty in a game progress with a degree according to the stored history of the violation of the rule when it is determined that the violation of the rule occurs.

[Detailed Description of the Invention]

[0001]

[Technical Field of the Invention]

The present invention relates to a game apparatus, method for advancing a game, program and recording medium in which a predetermined rule is applied in a game progress.

[0002]

[Prior Art]

In video games, generally, a player character is operated according to an instruction from a player and the game progresses based on the operation result of the player character. Various kinds of methods for advancing the game have been provided, and most of the video games include processing for a battle with a character that is operated by a video game apparatus or the other player in a game progress.

[0003]

Regardless of the genre, such as a role playing game, an action game, or an adventure game, there is known a game including a battle between the player character and an enemy character in the game progress. Moreover, for example, there is known a simulation game that causes a team of the player to battle an enemy team under the rule of sports to be simulated. In any kind of game, it is an important to maintain the player's interest in the game progress.

[0004]

Games which have a battle between a player character and an enemy character include a game that decides a character that will appear according to the circumstances of the actual game played by the player, and tries to maintain the player's interest by a battle with an enemy character with the appropriate amount of power (for example, refer to the Patent Document 1). In addition, there is a game in which a judge presents a yellow card or red card when a player's operation is against the rule as a sports simulation game (for example, refer to the Patent Document 2).

[0005]

[Patent Document 1]

Unexamined Japanese Patent Publication 2002-18142

[Patent Document 2]

Unexamined Japanese Patent Publication 2001-113050

[0006]

[Problems to be solved by the Invention]

However, in the game of the Patent Document 1, only the enemy character appearing is changed and the flow of the game progress is unchanged. In the game of the Patent Document 2, only the rule of the actual sports game is simulated, which is very common. In this way, in conventional games there were few games in which the player's instruction caused a novel change in the game progress itself. For this reason, it was difficult to cause the player to continue to have an interest in the game progress.

[0007]

An object of the present invention is to provide a game apparatus and the like that can cause a player to maintain interest in a game progress by providing a novel change in a flow of the game progress.

[0008]

[Means to Solve the Problems]

In order to achieve the above object, a game apparatus according to a first aspect of the present invention is a game apparatus that advances a game in accordance with a player's instruction, including progress degree detecting means for detecting a degree of a game progress, rule determining means for determining a rule applicable when the player advances the game according to the degree of the game progress detected by the progress degree detecting means, instruction input means for inputting a desired instruction of the player to advance the game, rule violation determining means for determining whether a violation of the rule determined by the rule determining means occurs based on the instruction input by the player from the instruction input means, and penalty processing means for imposing a predetermined penalty in the game progress when the rule violation determining means determines that the violation occurs.

[0009]

In the game apparatus according to the above first aspect, the rule violation determining means may include means that determines whether the instruction itself

input by the player from the instruction input means is against the rule determined by the rule determining means.

[0010]

The game apparatus according to the above first aspect may further include instruction executing means for executing processing according to the instruction input by the player from the instruction input means, and in the game apparatus, the rule violation determining means may include means that determines whether a processing result of the instruction executing means based on the instruction input by the player from the instruction input means is against the rule determined by the rule determining means.

[0011]

In the game apparatus according to the above first aspect, the rule as an object on which the penalty is imposed changes according to the game progress. This causes a novel change in the flow of the game progress, which makes it possible for the player to maintain interest in the game for a long time.

[0012]

In the game apparatus according to the above first aspect, the rule applicable when the game advances may be divided into a plurality of groups. In this case, the rule determining means can determine a rule that belongs to a group according to the degree of the game progress detected by the progress degree detecting means as the rule applicable when the game advances.

[0013]

In the game apparatus according to the above first aspect, the number of rules applicable when the game advances may be two or more. In this case, the rule determining means can increase the number of rules applicable when the game advances as the degree of the game progress detected by the progress degree detecting means advances.

[0014]

In the game apparatus according to the above first aspect, in the case where the instruction input means inputs an instruction to operate a player character, the game apparatus can further include characteristic value storage means for storing a characteristic value of the player character that varies according to circumstances of the game progress and exerts an influence on the game progress, and the penalty processing means can impose the predetermined penalty by varying the characteristic value of the player character stored in the characteristic value storage means.

[0015]

In the game apparatus according to the above first aspect, in the case where the instruction input means inputs a type of operation performed by the player character, which is allowed to perform a plurality of types of operations, the penalty processing means can impose the predetermined penalty by limiting the types of operations which the player character is allowed to perform.

[0016]

The number of player characters to which the operation is allowed to be instructed from the instruction input means may be two or more, and in this case, the rule violation determining means can include means that judges a player character that causes the violation of the rule determined by the rule determining means based on the instruction input from the instruction input means among the plurality of player characters, and the penalty processing means can impose the predetermined penalty on the player character that causes the violation of the rule.

[0017]

The game apparatus according to the above first aspect may further include item storage means for storing items given to the player with the game progress. In this case, the penalty processing means can impose the predetermined penalty by deleting a predetermined item among the items stored in the item storage means.

[0018]

At this point, the item storage means may store the items given to the player by classifying the items into groups determined based on a value in the game progress. In this case, the rule violation determining means can also determine a degree of the violation of the rule when the rule violation determining means determines that the violation of the rule determined by the rule determining means occurs based on the instruction input from the instruction input means, and the penalty processing means can delete an item which belongs to a group according to the degree of the violation of the rule among the items stored in the item storage means.

[0019]

In order to achieve the above object, a game apparatus according to a second aspect of the present invention is a game apparatus that advances a game in accordance with a player's instruction, including rule storage means for storing a rule applicable when the player advances the game, instruction input means for inputting a desired instruction of the player to advance the game, rule violation determining means for determining whether a violation of the rule stored in the rule storage means occurs based on the instruction input by the player from the instruction input means, violation history storage means for storing a history of violations determined by the rule violation determining means, and penalty processing means for imposing a predetermined penalty in a game progress with a degree according to the violation history stored in the violation history storage means when the rule violation determining means determines that the violation occurs.

[0020]

In the game apparatus according to the above second aspect, the rule violation determining means may include means that determines whether the instruction itself input by the player from the instruction input means is against the rule stored in the rule storage means.

[0021]

The game apparatus according to the above second aspect may further include instruction executing means for executing processing according to the instruction input by the player from the instruction input means, and in the game apparatus, the rule violation determining means may include means that determines whether a processing result of the instruction executing means based on the instruction input by the player from the instruction input means is against the rule stored in the rule storage means.

[0022]

In the game apparatus according to the above second aspect, the weight of the penalty to be imposed when the violation of the rule occurs is changed based on the history of the violations of the rule. This causes a novel change in the flow of the game progress, which makes it possible for the player to maintain interest in the game for a long time.

[0023]

In order to achieve the above object, a method for advancing a game according to a third aspect of the present invention is a method for advancing the game executed by a computer apparatus, including the steps of detecting a degree of a game progress, determining a rule applicable when a player advances the game according to the detected degree of the game progress, inputting a desired instruction of the player to advance the game, determining whether a violation of the determined rule occurs based on the instruction input by the player, and imposing a predetermined penalty in the game progress when it is determined that the violation of the rule occurs.

[0024]

In order to achieve the above object, a method for advancing a game according to a fourth aspect of the present invention is a method for advancing the game executed by a computer apparatus that stores a rule applicable when a player advances the game, including the steps of inputting a desired instruction of the player to advance the game,

determining whether a violation of the stored rule occurs based on the instruction input by the player, storing a history of the determined violations of the rule, and imposing a predetermined penalty in a game progress with a degree according to the stored history of the violation of the rule when it is determined that the violation of the rule occurs.

[0025]

In order to achieve the above object, a program according to a fifth aspect of the present invention is a program for advancing a game, which program is executed by a computer apparatus, allowing the computer apparatus to execute the steps of detecting a degree of a game progress, determining a rule applicable when a player advances the game according to the detected degree of the game progress, inputting a desired instruction of the player to advance the game, determining whether a violation of the determined rule occurs based on the instruction input by the player, and imposing a predetermined penalty in the game progress when it is determined that the violation of the rule occurs.

[0026]

In order to achieve the above object, a program according to a sixth aspect of the present invention is a program for advancing a game, which program is executed by a computer apparatus that stores a rule applicable when a player advances the game, allowing the computer apparatus to execute the steps of inputting a desired instruction of the player to advance the game, determining whether a violation of the stored rule occurs based on the instruction input by the player, storing a history of the determined violations of the rule, and imposing a predetermined penalty in a game progress with a degree according to the stored history of the violation of the rule when it is determined that the violation of the rule occurs.

[0027]

The computer apparatus in the method for advancing the game according to the above third and fourth aspects and in the program according to the fifth and sixth

aspects may be a general-purpose personal computer or the like, or a video game apparatus dedicated to the video game. The game apparatus according to the above first and second aspects can be applied to a general-purpose computer such as a personal computer in which each of the above means is constructed by the program, in addition to an apparatus dedicated to the game. In addition, the game apparatus can be applied to other electronic equipment, which can operate as a computer apparatus, such as a cellular phone. These game apparatuses may be used regardless of whether they are stationary types or portable types. The program according to the fifth and sixth aspects can be recorded on a computer-readable storage medium to be provided, and can be distributed via a network.

[0028]

[Embodiment of the Invention]

An embodiment of the present invention will be explained with reference to the accompanying drawings.

[0029]

FIG. 1 is a block diagram illustrating the configuration of a portable game device that is applied to this embodiment. As illustrated in the figure, this portable game device includes a game device main body 1 and a cartridge 2 which is inserted into a slot of the game device main body 1 and is individually provided for each game. The game according to this embodiment is implemented by inserting a specific thing as the cartridge 2 into the game device main body 1.

[0030]

The game device main body 1 includes a CPU (Central Processing Unit) 11, a boot ROM (Read Only Memory) 12, a work RAM (Random Access Memory) 13, a controller 14, an LCD (Liquid Crystal Display) 15, a speaker 16, and a communications port 17. The cartridge 2 has a ROM 21 and a RAM 22. The ROM 21 and RAM 22 of the cartridge 2 are connected to the CPU 11 of the game device main body 1 via a

connector 20.

[0031]

The respective structural components 12 to 17 of the game device main body 1 and the ROM 21 and RAM 22 of the cartridge 2 are connected to the CPU 11, and the CPU 11 executes programs stored in the boot ROM 12 and ROM 21 to control throughout the portable game device. The boot ROM 12 prestores a boot program for initializing the work RAM 13 of the game device main body 1 and a register of the CPU 11 and the like. The work RAM 13 is used as a work area when the CPU 11 executes the program.

[0032]

The controller 14 includes a cross key, which functions as a joy stick, and multiple operation keys, and inputs a player's instruction by these key operations. The LCD 15 displays a game space, a character existing therein and a message necessary for the game progress according to the execution of the game program by the CPU 11. The speaker 16 outputs sound effects and the like according to the circumstance of the game progress in accordance with the game program by the CPU 11. The communications port 17 transmits/receives information to/from another portable game device or a stationary game device.

[0033]

In the cartridge 2, the ROM 21 stores a game program for implementing the game according to this embodiment. When the game is interrupted by the player's instruction, the RAM 22 saves data relating to the circumstances of the game progress stored in the work RAM 13 in order to restart the game from the interrupting time. The RAM 22 is backed up by a battery, which is not shown in the figure, and the storage contents are not erased even if power of the game device main body 1 is interrupted and further the cartridge 2 is detached from the game main body 1. In place of the RAM 22, a writable/erasable nonvolatile memory such as a flash memory may be used.

[0034]

Next, the procedure of the game progress according to this embodiment will be briefly explained. In the process that progresses the game, some events occur, and the player clears the events to advance the game. The game is advanced as the player operates the player character, and the player can arbitrarily select a player character to be used in the game from multiple player characters. At an event occurring time, the player can select two or more player characters that participate in the event, and further choose one of the selected player characters as a leader.

[0035]

FIG. 2 is a view schematically illustrating an occurrence of the events in the game according to this embodiment. This game has a possibility that eleven events including event 1 to event 11 will occur before the game ends. Each event occurs when a predetermined condition fixed for each event is established. Events 1 to 6 among events 1 to 11 are those that are essential for the story of the game, and the story of the game does not progress unless these events are cleared. Every time when each essential event is cleared, a stage of the game progress proceeds. After the event 6 is cleared, an ending image is accompanied, thereafter the game ends.

[0036]

Events 7 to 11 are arbitrary events, and even if these events can be neither generated nor cleared, the story of the game can be moved forward. Events 7 to 9 among these are those that occur at random regardless of the stage of the game progress. Event 10 may occur between events 1 and 2 and event 11 may occur between events 3 and 4.

[0037]

In order to advance the game as clearing the aforementioned events 1 to 11, the player moves the player characters on the map. One that occurs as an event is a battle with the enemy character (in some cases, multiple enemy characters are provided and a

leader is fixed) in this embodiment. A screen to be displayed on the LCD 15 is changed at the time when the player characters are moved on the map during the events 1 to 11 by the battle and when the events 1 to 11 by the battle occur.

[0038]

FIG. 3(a) is a view illustrating a screen on which a world map is displayed. A world map 100 is formed in such a way that multiple towns (shown by circles in the figure) are arranged on a field having a predetermined geographic feature formed. A location where a player character 130 is displayed is the present location. The player character is moved between the towns on the world map by operation of the controller 14. The player character moves between the towns on the world map, so that a virtual, conceptual date in the game is updated.

[0039]

A local map, which is not shown in the figure, is formed for each town. The player operates the controller 14 to move the player character on the local map. Alternatively, the player character may move on only the world map 100 and a battle map 110 to be described later (refer to FIGS. 3 and 4) instead of the local map. If a predetermined condition is established while the player character moves on the world map 100 and the local map, the events by the battle occur.

[0040]

FIGS. 3(b) and (c) are views each illustrating a battle screen, and (b) illustrates a battle screen that is displayed when the battle is started, and (c) illustrates a victory condition display screen that is displayed at first in the battle. The battle between the player characters and the enemy characters is performed on the battle map 110.

[0041]

In the battle screen at the starting time shown in FIG. 3(b), an instruction selection window 120 and an instruction object character window 121 are displayed in addition to a leader 131 of the player characters. Although the instruction selection window 120

is used to select and input the player's instruction, the contents, which are different from this, may be displayed. When there is no player character as an object to which the player can input an instruction, the instruction object character window 121 is not displayed in some cases. Meanwhile, in the victory condition display screen illustrated in FIG 3(c), a message 122 indicating a condition for a victory and a leader 132 of the enemy characters are displayed.

[0042]

By the way, battle screens in FIGS. 3(b) and(c) illustrate the same battle. However, the portions of the battle map 110 displayed on the LCD 15 are different from each other.

[0043]

FIG 4 is a view explaining movement of a display portion in the battle screen. When the battle is started, the multiple characters including the leader 131 of the player characters and the leader 132 of the enemy characters are arranged on the battle map 110. In the battle screen at the starting time, an area 15b having a predetermined size where the leader 131 of the player characters is placed at the center is displayed as a battle screen at the starting time of FIG 3(b). The battle map 110 displayed in the battle screen at the starting time of FIG 3(b) corresponds to a portion included in an area 15b of the battle map 110 of FIG 4.

[0044]

Meanwhile, in the victory condition display screen, since the victory condition is "defeat the enemy leader", an area 15c (having the same size as that of the area 15b) where the leader 132 of the enemy characters is placed at the center is displayed as the victory condition display screen of FIG 3(c). The battle map 110 displayed in the victory condition display screen of FIG 3(c) corresponds to a portion included in the area 15c of the battle map 110 of FIG 4.

[0045]

In FIG. 3(c), since the victory condition is “defeat the enemy leader”, the area 15c where the leader 132 of the enemy characters is placed at the center is displayed as the victory condition display screen, but in the case of another victory condition, the victory condition display screen is not always decided in this way. For example, if the victory condition is “move to designated point”, an area around the designated point can be displayed as the victory condition display screen.

[0046]

Although not illustrated, in the battle screen where the battle is in progress, the area to be displayed on the LCD 15 is sequentially changed such that the player can easily recognize the circumstances of the battle progress. In the battle screen at the starting time, the victory condition display screen and the battle screen where the battle is in progress, the reason why the portion to be displayed is thus moved is to balance the size of display and the contents since the display area of the LCD 15 of the portable game device is limited. To explain it more specifically, the reason why the portion to be displayed is thus moved is to display each character with such a size that the player can recognize the characters in the screen with the limited size without considering the size of the battle map 110 too much.

[0047]

In the battle, an operation that the player should not instruct to the player character, and an operation in which the result obtained by instructing the operation should not be a predetermined result are defined as rules in the game. The rules change according to the circumstances of the game progress, but when a rule violation occurs, a yellow card or red card is presented, and a penalty is imposed on the player. The rules and the penalty to be imposed in the case of the rule violation will be specifically described later.

[0048]

An explanation will next be given of various data to be used in the game according

to this embodiment. Variable data, which varies according to the game progress, is stored in the work RAM 13 of the game main body 1, and fixed data, which does not vary according to the game progress, is prestored in the ROM 21 of the cartridge 2. As for variable data, when the game is started from the beginning, initial data stored in the ROM 21 is transferred to the work RAM 13. When the game is restarted midway, data saved in the RAM 22 is transferred to the work RAM 13.

[0049]

FIG. 5 is a view illustrating variable data to be stored in the work RAM 13. Though there are various kinds of variable data necessary for the game progress, the following will explain data relating to the present invention.

[0050]

FIG. 5(a) shows current data that is changed according to the game progress, namely, data indicating currently under what circumstances the game is advanced. This current data includes general data 201 and player character-specific data 202-1 to 202-n.

[0051]

General data 201 is common data regardless of data by player character, and includes Gil (amount of virtual money that the player possesses in the world of the game) and Item (name of the item that the player possesses), and Item_num (the number of items). When the number of kinds of items that the player possesses is two or more, combinations of Item and Item_num, which correspond to the number of kinds, are provided and registered. The items, which the player can possess, are classified into light items and heavy items according to their values in the game progress, and they are registered as Item data by classified group.

[0052]

The player character-specific data 202-1 to 202-n include Name (name of the player character), Indiv (player character unique ID), Level (current level of the player

character), Exp (Experience value: one that determines a level), HP (Hit Point: amount of residual damage withstanding the attack from the enemy, the battle is impossible when 0), HPMAX (Hit Point MAXimum value), MP (Magic Point: amount of usable magic), MPMAX (Magic Point MAXimum value), Status (there is a case where the battle is impossible), StrAttack (physical attack), StrDefense (physical defense), IntAttack (magic attack), IntDefense (magic defense), Speed (speed that decides an attack order in the battle), Flag_Battle (Flag in Battle), Flag_Prison (Flag in Prison (battle participation impossible) flag), StopBattle (the number of residual battles in prison), Xpos (X-coordinate on the battle map), Zpos (Z-coordinate on the battle map), Redcard (the cumulative number of red cards), Yellowcard (the cumulative number of red cards), and netID (ID of the character in battle that is allocated at the time of participating in the battle).

[0053]

Among these data, Level, HP, HPMAX, MP, MPMAX, Status, StrAttack, StrDefense, IntAttack, IntDefense, and Speed indicate whether the player character has an advantageous position or a disadvantageous position in the battle, namely they are an example of characteristic values of the player character that exert a direct influence on the game progress.

[0054]

Regarding the enemy character, though not illustrated, data is prepared in the ROM 21 like the player character, and when the enemy character participates in the battle, the data is copied to the work RAM 13 and used. The data of the enemy character includes Name, Indiv, Level, Exp, HP, HPMAX, MP, MPMAX, Status, StrAttack, StrDefense, IntAttack, IntDefense, Speed, Xpos, and Zpos. The meanings of these data are the same as those of data corresponding to the player character.

[0055]

FIG. 5(b) is a view illustrating law check data. The law is selected and applied as

a rule at each point in the game progress. Both the rule and the law are regulations relating to the game progress, and in this embodiment, the law is applicable as a regulation, and the rule is actually applied as a regulation in the event. Law check data in FIG. 5(b) includes general data 211 and player character-specific data 212-1 to 212-n.

[0056]

The general data 211 includes MenuID (menu that is designated in connection with the action of the player character at the last minute), MenuID_OLD (menu that is recorded on MenuID at the last minute), AbilityID (ability ID when the designated menu is ability), and AbilityID_OLD (ability ID that is recorded on Ability ID at the last minute).

[0057]

The player character-specific data 212-1 to 212-n include TermFlag (indicating that another character is made impossible to battle by petrifying the other character or HP = 0 is established), ItemFlag (indicating that an item is stolen from the other character), Status (indicating that a status abnormality is provided to the other character), TargetNum (the number of characters as a target for an action), Damage [1]-[k] (damage or recovery value given to the other character by the action), Weapon [1], [2] (kinds of weapons used in the action), CardFlag [1], [2] (indicating that a yellow card or red card is received in the battle), and LawID [1], [2] (law unique ID of the law that has been violated (to be described later)).

[0058]

FIG. 6 is a view illustrating a law table in fixed data stored in the ROM 21. The law table includes two tables, that is, an individual law table illustrated in FIG. 6(a) and a staged law table illustrated in FIG. 6(b).

[0059]

In the individual law table illustrated in FIG. 6(a), a law unique ID, a rule name, a judgment type, an offset, a red card condition, and a penalty type are stored in

association with each other. The law unique ID is an ID for identifying each law. The law name is a name that is given to each law. The judgment type is used to judge whether the action of the character is against the rule. The offset shows a specific action or a result of the action that is against the rule to be used for judgment. The red card condition shows a condition when the red card is presented instead of the yellow card even at the first rule violation. The penalty type shows the type of the penalty that is imposed on the player in the case of the rule violation.

[0060]

In the staged law table in FIG. 6(b), a staged law ID and a law unique ID for each stage are stored in association with each other. The staged law ID corresponds to the date of each stage. In the case of the first day of first stage 1, a law, which includes a staged law ID of "001" and a law unique ID of "008" (namely, damage of 20 or more), is applied as a rule.

[0061]

An explanation will be next given of the penalty to be given to the player when the rule violation occurs according to the law tables in FIG. 6. FIG. 7 is a view illustrating penalty types and the contents. This figure is not stored in the ROM 21 as a table, but a game program is created such that the contents described here are performed as the penalty.

[0062]

The penalty types include a fine (LOST_GIL), a status down (STAT_DOWN), and a consumption item confiscation (LOST_TREASURE), and as shown in the figure, the weight of the penalty differs depending on whether the yellow card is presented by the rule violation or the red card is presented. In a case that the penalty type is the fine or the consumption item confiscation, the penalty is imposed on the entire player character as an object to be applied. In a case that the penalty type is the status down, the penalty which degrades any one or two kinds of, for example, Level, HP, HP MAX, MP,

MPMAX, Status, StrAttack, StrDefense, IntAttack, IntDefense, and Speed, is imposed on the player character that violates the rule.

[0063]

The following will explain processing to be executed to advance the game in the game according to this embodiment. When processing other than display processing is carried out, there is a case in which processing for displaying an image corresponding to the processing on the LCD 15 is carried out. However, regarding display processing other than display processing unique to this embodiment, some explanations will be omitted.

[0064]

When power of the game device main body 1 is turned on, an initial screen for selecting whether the game is started from the beginning or restarted midway is displayed on the LCD 15. The player operates the controller 14 according to the initial screen to select whether the game is started from the beginning or restarted midway. When the game is selected to start from the beginning, initial data is loaded on the work RAM 13 from the ROM 21 and processing of the game is started, and when the game is selected to restart midway, saved data is loaded on the work RAM 13 from the RAM 22 and processing of the game is started.

[0065]

FIG. 8 is a flowchart illustrating processing of a main routine in the game according to this embodiment. When the processing of the main routine is started, the CPU 11 determines whether a player's instruction is input from the controller 14 (step S101). Before the player's instruction is input from the controller 14, processing in step S101 is repeated to wait for inputting of some kind of instruction from the player.

[0066]

When the player's instruction is input, the CPU 11 determines whether the input instruction is a final instruction (step S102). When the input instruction is not the final

instruction, the CPU 11 determines whether the input instruction is a movement instruction to the player character (step S103). When the input instruction is not the movement instruction, the CPU 11 performs processing according to the contents of the instruction (step S104) and the processing flow proceeds to step S108.

[0067]

When the input instruction is the movement instruction, the CPU 11 moves the player character on the map (the world map 100 or the local map) (step S105). The CPU 11 determines whether the player character has moved on the world map 100 (step S106). When the player character has moved on the world map 100, the CPU 11 updates a date in the game, and sets the law corresponding to the updated date in the current stage as the rule (step S107). After that, the processing flow proceeds to step S108.

[0068]

In step S108, the CPU 11 determines whether an event occurrence condition has been established. When the event occurrence condition has been established, the CPU 11 causes the event whose occurrence condition has been established to occur and executes event processing that shifts to processing relating to the event (step S109). The details on the event processing will be described later. When the event is ended in the event processing, the CPU 11 determines whether the ended event is any one of events 1 to 5 (step S110).

[0069]

When the ended event is any one of events 1 to 5, the CPU 11 changes the setting of the game stage to a next stage (step S111). Then, the processing flow goes back to step S101. When the event occurrence condition is not established in step S108 or the ended event is not any one of events 1 to 5 in step 110, the processing flow goes back to step S101 directly.

[0070]

When the instruction input in step S102 is the end instruction, the CPU 11 performs save processing for saving data of the game stored in the work RAM 13 to the RAM 22 of the cartridge 2 (step S112). When the save processing is ended, the processing in this flowchart is ended.

[0071]

The following will specifically explain the event processing in step S109. FIG. 9 is a flowchart specifically illustrating the event processing. An event in this event processing is the battle between the player characters and the enemy characters as mentioned above, and when the event processing is started, the CPU 11 first receives a player's instruction input from the controller 14 in connection with the selection of the player characters that participate in the battle (step S201).

[0072]

When receiving the instruction about the selection of the player character, the CPU 11 places the selected player character and the enemy character that participates in the battle on the battle map 110, and performs processing for displaying them on the LCD 15 (step S202). The battle screen displayed here is the same as the battle screen at the battle starting time where the leader of the player characters is placed at the center as illustrated in FIG. 3(b). When this screen is once displayed, the CPU 11 performs victory condition display processing for displaying the victory condition display screen as illustrated in, for example, FIG. 3(c) on the LCD 15 (step S203).

[0073]

FIG. 10 is a flowchart specifically illustrating the victory condition display processing in step S203. In the victory condition display processing, the CPU 11 checks which victory condition is set in this battle (step S301). As a settable victory condition, there are provided five conditions, "defeat the enemy leader", "defeat all", "withstand for a fixed turn", "break the designated object", and "move to the designated point." The CPU 11 executes any one of condition 1 display processing to condition 5

display processing in steps S302 to S306 according to the check result of the victory condition.

[0074]

To explain the condition 1 display processing, in this case, the leader of the enemy characters is also placed on the battle map 110. The screen first displayed on the LCD 15 uses the area 15b as the display area where the leader 131 of the player characters is placed at the center as illustrated in FIG. 4, but the screen using the area 15b as the display area where the leader 132 of the enemy characters is placed at the center is displayed on the LCD 15.

[0075]

In the condition 2 display processing to the condition 5 display processing, similar to the leader 131 of the enemy characters in the condition 1 display processing, an object to be displayed at the center of the screen is fixed. One that is displayed at the center of the screen is the leader of the enemy characters in the condition 2 display processing, the leader of the player characters in the condition 3 display processing, the designated object in the condition 4 display processing, and the designated point in the condition 5 display processing, respectively.

[0076]

In any of the condition 1 display processing to the condition 5 display processing, an image including the battle map 110 other than the unit relating to the victory condition is toned down and displayed. The unit that is not toned down is the leader of the enemy characters in the condition 1 display processing, all enemy characters in the condition 2 display processing, all player characters in the condition 3 display processing, the designated object in the condition 4 display processing, and the designated point in the condition 5 display processing, respectively.

[0077]

When the victory condition display screen is displayed on the LCD 15 by any one

of the condition display processing in steps S302 to S306, the CPU 11 waits for a passage of a fixed time since the display of the victory condition display screen (step S307). After passing the fixed time since the display of the victory condition display screen, the CPU 11 returns the screen displayed on the LCD 15 to the original screen (screen displayed in step S202) (step S308). Then, the victory condition display processing is ended.

[0078]

When the victory condition display processing is ended and the processing flow goes back to the event processing in FIG. 9, the CPU 11 places a judge character on the battle map 110, and displays the battle screen including the judge character on the LCD 15 (step S204). This judge character is a non-player character that is used to present the yellow card or red card when the rule violation occurs. When the display of the judge character is ended, the CPU 11 executes battle processing for actually carrying out the battle between the player characters and the enemy characters (step S205). Regarding the battle screen since the battle processing is started, a displayed portion of the battle map 110 changes according to the circumstances of the battle progress.

[0079]

FIG. 11 is a flowchart specifically illustrating the battle processing in step S205. In the battle processing, the CPU 11 first performs initialization processing for initializing general data 211 and CardFlag and LawID of the player character-specific data 212-1 to 212-n of the law check data (step S401).

[0080]

Next, the CPU 11 determines a character that can act according to Speed among current data of the player character that participates in the battle and current data of the enemy character (step S402). When the character which can act is determined, the CPU 11 performs each action initialization processing for initializing player character-specific data 212-1 to 212-n of law check data (note that CardFlag and LawID

are excluded) (step S403).

[0081]

Next, the CPU 11 determines whether the character determined which can act in step S402 is the player character or the enemy character (step S404). When the character is the player character, the CPU 11 receives a player's instruction of the action to the player character from the controller 14 (step S405). Then, the processing flow proceeds to step S407.

[0082]

When the character is an enemy character, the CPU 11 determines an action that the enemy character should perform according to data relating to the enemy character. In this case, an action of the enemy character such as to cause the rule violation by the law set in step S107 is not determined (step S406). When the action of the enemy character is determined, the processing flow proceeds to step S407.

[0083]

In step S407, the CPU 11 causes the player character or the enemy character determined which can act in step S402 to carry out the action instructed in step S405 or the action determined in step S406. The CPU 11 determines whether there is an unprocessed character among other characters as the targets in connection with the action of the character in step S407 (step S408).

[0084]

When there is an unprocessed character, the CPU 11 causes the character as the unprocessed target to perform an action subjected to an influence by the action of the character in step S407. This action includes not only just a passive action but also an active action that launches a counterattack against the character that launched an attack (step S409). After that, the processing flow goes back to step S408 again. When there is no unprocessed character in step S408, the CPU 11 performs action result record processing for recording the action of the character in step S407 and the action result

(step S410).

[0085]

FIG. 12 is a flowchart specifically illustrating the action result record processing in step S410. In the action result record processing, the CPU 11 first backs up the latest action. That is, the CPU 11 copies data of MenuID included in the general data 211 of law check data onto MenuID_OLD, and copies data of AbilityID onto AbilityID_OLD (step S501).

[0086]

The CPU 11 next records the action in step S407. That is, the CPU 11 records an action menu on the MenuID, and an Ability ID on the AbilityID (in a case where the A ability is selected). In a case where an active action, such as “fight,” is not instructed, but rather an instruction that substantially prevents the character from acting, such as “wait,” occurs, the CPU 11 clears data of the MenuID and AbilityID (step S502).

[0087]

The CPU 11 records a category of the weapon that is used by the character in the action in step S407. That is, when the action that uses the weapon such as “fight” is recorded on the MenuID, the category of the weapon, which the character has, is recorded on Weapon[2]. When A ability is recorded on the AbilityID, the CPU 11 checks the essential weapon in the A ability and records the category on the Weapon[2] (step S503).

[0088]

The CPU 11 further records the result of the actions in steps S407 and S409. That is, when even one enemy character is petrified or HP=0 is established due to the relevant action, the CPU 11 turns on TermFlag of player character-specific data 212-1 to 212-n about the character that corresponds to the law check data. When the character that corresponds to the law check data steals an item from the enemy character due to the relevant action, the CPU 11 turns on ItemFlag of player character-specific data

212-1 to 212-n about the character that corresponds to the law check data. When a status change occurs due to the relevant action, the CPU 11 records on Status of the player character-specific data 212-1 to 212-n about the character that corresponds to the law check data. The CPU 11 records the number of characters as targets due to the relevant action on TargetNum of the player character-specific data 212-1 to 212-n about the character that corresponds to the law check data. The CPU 11 records damage given to the other character due to the relevant action or a recovery value on Damage[1] to [k] of the player character-specific data 212-1 to 212-n about the character that corresponds to the law check data (step S504). Then, the action result record processing is ended.

[0089]

When the action result record processing is ended and the processing flow goes back to the battle processing in FIG. 11, the CPU 11 performs penalty check processing for checking whether a penalty is imposed on the player when the violation of the rule by the law set in step S107 occurs by the action of the character in stepS407 and the action result (step S411).

[0090]

FIG. 13 is a flowchart specifically illustrating the penalty check processing in step S411. In the penalty check processing, the CPU 11 checks whether there are player characters that are not yet processing objects to be subjected to the penalty check among the player characters, which are selected by the player in step S201 and which have participated in the battle (step S601).

[0091]

When there are player characters that are not yet processing objects, the CPU 11 determines whether HP of the player character as the processing object is 0 with reference to the player character-specific data 212-1 to 212-n of the current data in connection with any one of the player characters that are not yet processing objects (step

S602). When HP is not 0, the CPU 11 determines whether the relevant player character is impossible to battle by petrifying (step S603).

[0092]

When HP of the player character as a processing object is 0 or the player character as a processing object is impossible to battle by petrifying, processing that is performed to the relevant character is ended and the processing flow goes back to step S601. When the player character as a processing object is not petrified, the CPU 11 checks the determination type of the law set as a rule in step S107 (step S604).

[0093]

When the determination type of the law is Category, the CPU 11 determines whether data of MenuID in the general data 211 of the law check data is offset data of the law set as a rule in step S107 (step S605). When data of MenuID is not offset data, the processing flow goes back to step S601 directly. When data of MenuID is offset data, the CPU 11 performs card processing to be specifically described later (step S606). After returning from the card processing, the processing flow goes back to step S601.

[0094]

When the determination type of the law is Ability, the CPU 11 determines whether data of AbilityID in the general data 211 of the law check data is offset data of the law set as a rule in step S107 (step S607). When data of AbilityID is not offset data, the processing flow goes back to step S601 directly. When data of AbilityID is offset data, the CPU 11 performs card processing to be specifically described later (step S608). After returning from the card processing, the processing flow goes back to step S601.

[0095]

When the determination type of the law is Arm, the CPU 11 determines whether data of either Weapon [1] or [2] included in the player character-specific data 212-1 to 212-n of the law check data is offset data of the law set as a rule in step S107 (step S609). When neither Weapon [1] nor [2] is offset data, the processing flow goes back

to step S601 directly. When data of either Weapon [1] or [2] is offset data, the CPU 11 performs card processing to be specifically described later (step S610). After returning from the card processing, the processing flow goes back to step S601.

[0096]

When the determination type of the law is Damage, the CPU 11 determines whether a value, which is shown by data of any one of Damage [1] to [n] included in the player character-specific data 212-1 to 212-n of the law check data, is more than offset data of the law set as a rule in step S107 (step S611). When the value, which is shown by data of any one of Damage [1] to [n], is not more than offset data of the law, the processing flow goes back to step S601 directly. When the value, which is shown by data of any one of Damage [1] to [k], is more than offset data of the law, the CPU 11 performs card processing to be specifically described later (step S612). After returning from the card processing, the processing flow goes back to step S601.

[0097]

In step S601, when it is determined that there is no player character that is not yet processing object, the penalty check processing ends.

[0098]

FIG. 14 is a flowchart specifically illustrating card processing performed in step S606, S608, S610, or S612. In the card processing, the CPU 11 first the CPU 11 determines whether the red card condition of the law set as a rule in step S107 is STONE_TERM (step S701). When the red card condition is STONE_TERM, the CPU 11 further determines whether TermFlag included in the player character-specific data 212-1 to 212-n as processing objects is turned on.

[0099]

When TermFlag is turned on, the CPU 11 sets a flag, which indicates the red card, to either CardFlag [1] or [2], whichever is empty, included in the character-specific data 212-1 to 212-n of the relevant player character, and thereby records that the red card is

presented to the relevant player character because of the rule violation. The CPU 11 also adds a RedCard value included in current data about the relevant character by one, and thereby updates the cumulative number of red cards (step S703). Then, the processing flow proceeds to step S707.

[0100]

When the red card condition is not STONE_TERM or TermFlag is not turned on, the CPU 11 sets a flag, which indicates the yellow card, to CardFlag [1] included in the character-specific data 212-1 to 212-n of the relevant player character, or sets a flag, which indicates the yellow card, to CardFlag [2] when the flag is already set to CardFlag[1], and thereby records that the yellow card is presented to the relevant player character because of the rule violation. The CPU 11 also increases a Yellow Card value included in current data about the relevant character by one, and thereby updates the cumulative number of yellow cards (step S704).

[0101]

The CPU 11 performs yellow card display processing in which the judge character presents the yellow card to the relevant player character to which the yellow card is to be presented (step S705). The CPU 11 next checks whether both CardFlags [1] and [2] are turned on, and thereby determines whether a second yellow card is presented (step S706). When only the first yellow card is presented, the card processing ends and the processing flow goes back to penalty check processing in FIG. 13. When the second yellow card is presented, with the result that the red card is presented, the processing flow proceeds to step S707.

[0102]

The CPU 11 performs red card display processing in which the judge character presents the red card to the relevant player character to which the red card is to be presented (step S707). The CPU 11 next performs processing for causing the player character to which the red card is presented to leave the battle map 110 (step S708).

Then, the card processing is ended and the processing flow goes back to penalty check processing in FIG. 13.

[0103]

When the penalty check processing is ended and the processing flow goes back to the battle processing in FIG. 11, the CPU 11 determines whether a battle processing end condition is established (step S412). The battle processing end condition includes a stop condition for a predetermined reason or the like in addition to a victory condition and a defeat condition. When the battle processing end condition is not established, the processing flow goes back to the processing in step S402. Meanwhile, when the battle processing end condition is established, the battle processing ends.

[0104]

When the battle processing is ended and the processing flow goes back to the event processing in FIG. 9, the CPU 11 performs penalty processing for imposing a penalty on the player according to the passage of battle processing and the result thereof (step S206).

[0105]

FIG. 15 is a flowchart specifically illustrating the penalty processing in step S206. In the penalty processing, the CPU 11 determines whether there are player characters that are not yet processing objects to be subjected to the penalty among the player characters which are selected by the player in step S201 and which have participated in the battle (step S801).

[0106]

When there are player characters that are not yet processing objects, the CPU 11 determines whether there is a rule violation where no processing is performed in connection with any one of the player characters that are not yet processing objects (step S802). When there is no rule violation where no processing is performed in connection with the player character as the processing object, the processing flow goes

back to the step 801, and the CPU 11 further determines whether there is any further player character that is not yet a processing object.

[0107]

When there is a rule violation where no processing is performed, the CPU 11 detects the penalty type of the law set as the rule in step S107 (step S803). The CPU 11 also determines whether there is a rule violation in which the red card is presented with reference to CardFlag[1], [2] in connection with the player characters as processing objects (step S804).

[0108]

When there is a rule violation where the red card is presented, the CPU 11 imposes a penalty defined for the red card in connection with the penalty type detected in step S803. In particular, when the penalty type is STAT_DOWN, the CPU 11 decreases two kinds of statuses (for example, HP and MP) in connection with the player character as a processing object here (namely, the player character that is against the rule) (step S805).

[0109]

The CPU 11 further calculates the number of battle stop times where the relevant player character cannot participate in the battle because of the red card (step S806). The number of battle stop times can be obtained by an equation, for example, $((\text{the cumulative number of yellow cards}) + (\text{the cumulative number of red cards}) \times 2) \times 0.2 + 1$. The CPU 11 sets the calculated number of battle stops to data of StopBattle about the relevant player character (step S807). Then, the penalty processing is ended.

[0110]

Meanwhile, even though there is a violation of rule, when no rule violation deserving the red card occurs, the CPU 11 imposes a penalty defined for the yellow card about the penalty type detected in step S803. In particular,, when the penalty type is STAT_DOWN, the CPU 11 decreases one kind of status in connection with the player

character as a processing object here (namely, the player character that is against the rule) (step S808). Then, the penalty processing is ended.

[0111]

When the battle processing is ended and the processing flow goes back to the event processing in FIG. 9, the CPU 11 performs a predetermined end processing (step S207). This ends the event processing, and the processing flow returns to the aforementioned main routine in FIG. 8.

[0112]

As explained above, in the game according to this embodiment, the stage is updated by the end of the event and the date is updated by movement of the player character on the world map 100, and thereby the law set as the rule in the battle is changed. That is, since the rule changes sequentially according to the degree of the game progress, novel change occurs in the flow of the game progress to make it possible to continue to hold the player's interest.

[0113]

When there is a rule violation, a penalty is imposed on the player. The multiple kinds of penalties are prepared and changes according to the set law. This causes a novel change in the flow of the game progress and makes it possible to maintain the player's interest.

[0114]

The yellow card or red card is presented by the degree (contents or the number of times) of the rule violation, and the weight of the penalty to be imposed on the player changes depending on which card is presented. When the red card is presented, the number of battle stop times is changed by the cumulative number of yellow cards or red cards. This also causes a novel change in the flow of the game progress.

[0115]

Multiple player characters can be selected as the player characters participating in

the battle by the player's instruction, but when multiple player characters are selected, there is a case that a penalty is imposed on only the player character that breaks the rule (case in which the penalty type is status down). Meanwhile, when the penalty type is a fine or a consumption item confiscation, the penalty is imposed as a whole. Since there is a wide range of variations in the penalty type as above, influence on the game progress varies by the penalty.

[0116]

By the way, there is not only a case where it is determined to be the rule violation due to the player's instruction on the action but also a case where it is first determined to be the rule violation due to the result of the action instructed by the player. Since the latter includes an uncertain element in which it is uncertain that the rule violation occurs unless the player character is caused to perform an action, the change caused in the game progress varies.

[0117]

The present invention is not limited to the aforementioned embodiment and various modifications and applications may be possible. The following will explain some modifications of the aforementioned embodiment that are applicable to the present invention.

[0118]

In the aforementioned embodiment, there was a case in which the action itself which the player instructed as the action of the player character from the controller 14 was the rule violation and a case in which the result of the action was the rule violation, but the action instructed by the player was performed regardless of whether it was the rule violation or not. In contrast to this, when the action instructed by the player is the rule violation, the presentation of the yellow card and the record of the rule violation may be performed at the time and the instructed action may not be actually performed.

[0119]

In the aforementioned embodiment, the stage of the game was updated when the events 1 to 5 were ended, and the date in the game was updated when the player character moved on the world map 100. Then, the law set as the rule changed as the stage of the game and the date advanced based on the staged law table in FIG. 6(b). The law set as the rule at the same time was only one. However, the present invention is not limited to this.

[0120]

The change in the degree of the game progress, which exerts an influence on the change in the rule, may be based on a change in, for example, an experience value of the player character or the like instead of the change in the stage due to the end of the event and the change of the date. Moreover, the multiple laws are simultaneously set as a rule, and thereby the multiple rules may be simultaneously applied in the battle. In a case where the multiple laws can be simultaneously set as a rule, the number of laws set as a rule is increased as the degree of the game progress advances, and the number of rules to be simultaneously applied to the battle may be increased.

[0121]

Furthermore, the multiple laws are classified into groups and the group of laws set as rules may be decided from multiple groups according to the degree of the game progress. In this case, when the group including the multiple laws is set as a rule, each law belonging to the relevant group is set as the rule, and therefore there are multiple rules to be simultaneously applied to the battle.

[0122]

In the aforementioned embodiment, when there was the rule violation, three consequences, namely, the fine, the status down and the consumption item confiscation were defined as a penalty imposed on the player. However, the penalties to be imposed were not limited to these. For example, the kinds of weapons that can be used when the player character performs physical attack by the instruction from the

player may be limited (for example, prohibition against taking up a bow and arrow and the like) and the kinds of magic that can be used may be limited (for example, prohibition against working recovery magic and the like). A predetermined status abnormality may be caused on the player character. Instead of the status down of the player character, the status up may be caused on the enemy character.

[0123]

In the aforementioned embodiment, the weight of the penalty imposed on the player when the rule violation occurred was changed by depending on whether only the yellow card was presented because of the rule violation or the red card was also presented. However, the weight of the penalty imposed on the player can be changed by another method. For example, in a case where the status down is set as a penalty, the degree of status down, which is one kind of set penalty, may be changed. Not only the law applied as the rule but also the weight of penalty can be changed according to the degree of the game progress.

[0124]

The penalty weight may be changed according to the cumulative number of yellow cards (Yellowcard) and the cumulative number of red cards (Redcard) recorded on the player character-specific data 202-1 to 202-n of current data. For example, a violation historical index is obtained similar to the number of battle stop times obtained in step S806, and the degree of penalty imposed on the player may be increased more with an increase in the violation historical index. This further causes a novel change in the game progress to make it possible to continue holding the player's interest in the game.

[0125]

Furthermore, the degree of penalty imposed on the player may be changed by the total of the cumulative number of yellow cards and the cumulative number of red cards regarding all player characters instead of the cumulative number of player character-specific yellow cards and the cumulative number of player character-specific

red cards.

[0126]

In the aforementioned embodiment, the player character performed the action that was against the rule by the player's instruction, but the enemy character was controlled not to perform the action that was against the rule. In contrast to this, the enemy character can perform the action that is against the rule, and the yellow card or red card may be presented to the enemy character based on the action that is against the rule to be imposed the same penalty as presented to the player character. The action that is against the rule may be made different between the player character and the enemy character.

[0127]

In the aforementioned embodiment, the portable game device including the game device main body 1 and the cartridge 2 was used as a game device serving as a platform where the game of the present invention was executed. However, the game of the present invention may be executed using a stationary game device and a general-purpose computer such as a personal computer as a platform. Furthermore, the game of the present invention may be executed using other electronic equipment as a platform, which can function as a computer apparatus such as a cellular phone.

[0128]

In the aforementioned embodiment, the program for executing the video game of the present invention was stored in the ROM 21 of the cartridge 2 and distributed, but the program may be stored in a storage medium such as a CD-ROM, a DVD-ROM and the like, and distributed. In contrast to this, this program may be stored in a fixed disk device, which a server apparatus existing in the Internet has, and distributed to the game device main body 1 in FIG. 1 via the Internet. The program distributed from the server apparatus can be stored in a flash memory and the like provided in the game device main body 1 or the cartridge 2 and executed therefrom.

[0129]

[Advantageous Effect of the Invention]

As explained heretofore, according to the present invention, the rule as an object on which the penalty is imposed changes according to the game progress, resulting in a novel change in the flow of the game progress, which makes it possible for the player to maintain interest in the game for a long time.

[0130]

Also, when the weight of the penalty to be imposed in a case where the rule violation occurs is changed based on the history of the rule violation, a novel change is caused in the flow of the game progress, which makes it possible for the player to maintain interest in the game for a long time.

[Brief Description of the Drawings]

[FIG. 1]

It is a block diagram illustrating the configuration of a portable game device that is applied to an embodiment of the present invention.

[FIG. 2]

It is a view schematically illustrating occurrence of an event in the game according to an embodiment of the present invention.

[FIG. 3]

It is a view illustrating a world map and a battle screen.

[FIG. 4]

It is a view explaining movement of a display portion in the battle screen.

[FIG. 5]

It is a view illustrating variable data stored in a work RAM.

[FIG. 6]

It is a view illustrating a law table among fixed data stored in a ROM of a cartridge.

[FIG. 7]

It is a view illustrating penalty types and the contents.

[FIG. 8]

It is a flowchart illustrating processing of a main routine in the game according to an embodiment of the present invention.

[FIG. 9]

It is a flowchart specifically illustrating event processing of FIG. 8.

[FIG. 10]

It is a flowchart specifically illustrating victory condition display processing of FIG. 9.

[FIG. 11]

It is a flowchart specifically illustrating battle processing of FIG. 9.

[FIG. 12]

It is a flowchart specifically illustrating action result record processing of FIG. 11.

[FIG. 13]

It is a flowchart specifically illustrating penalty check processing of FIG. 11.

[FIG. 14]

It is a flowchart specifically illustrating card processing of FIG. 13.

[FIG. 15]

It is a flowchart specifically illustrating penalty processing of FIG. 9.

[Explanation of Referenced Numerals]

- 1 game device main body
- 2 cartridge
- 11 CPU
- 12 boot ROM
- 13 work RAM
- 14 controller

- 15 LCD
- 16 speaker
- 17 communications port
- 21 ROM
- 22 RAM

[Document Name] ABSTRACT

[Summary]

[Problems to be solved]

A novel change in a flow of a game progress causes a player to maintain interest in the game progress.

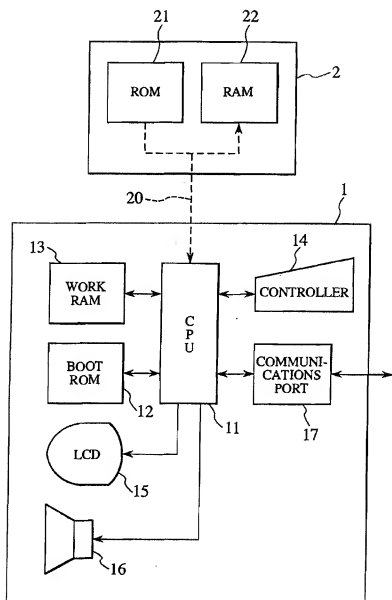
[Means to Solve the Problems]

A rule (applied to an event by a battle) is set according to a degree of the game progress every time a stage of the game advances. When the event by the battle occurs, and the battle is actually started, either a player character or an enemy character that can execute an action is determined. When the player character can execute the action, a player instructs the action. It is recorded whether the action instructed by the player or a result of the action is against the set rule. After the battle ends, a predetermined penalty is imposed on the player based on the recorded rule violation.

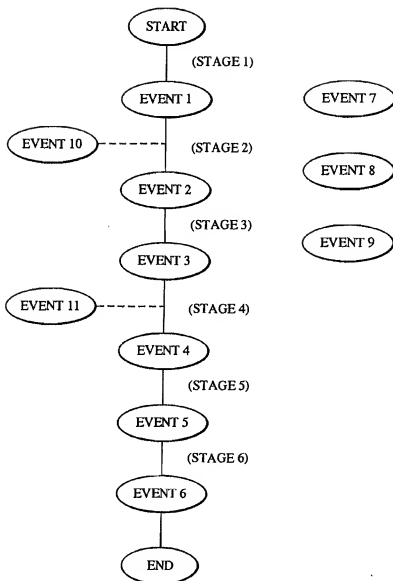
[Selected Figure] FIG. 9

[DOCUMENT NAME] DRAWINGS

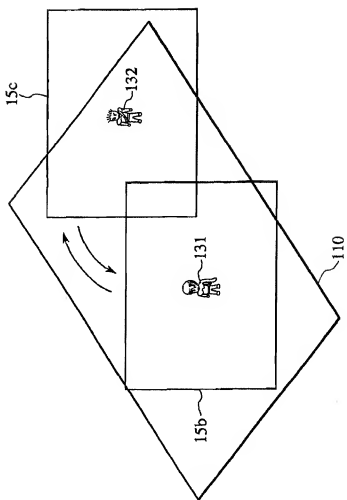
[FIG. 1]



[FIG. 2]



[FIG. 4]



[FIG. 3]

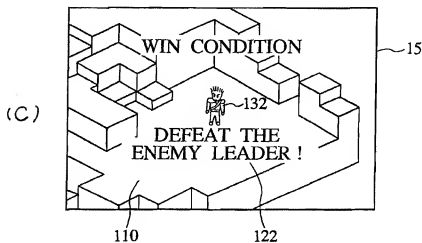
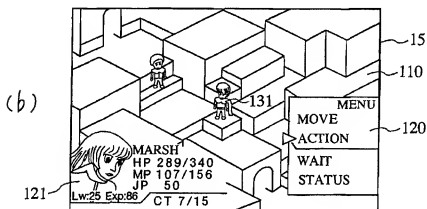
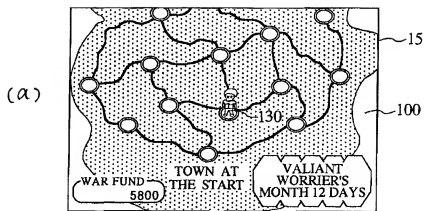


FIG. 5J

(a)

Gil	201
Item[1]	
Item_num[1]	
⋮	
Name	202-1
Indiv	
Level	
Exp	
HP	
HPMAX	
MP	
MPMAX	
Status	
StrAttack	
StrDefense	
IntAttack	
IntDefense	
Speed	
Flag_Battle	202-2
Flag_Prison	
StopBattle	
Xpos	
Zpos	202-3
Redcard	
Yellowcard	
netID	
Name	202-n
Indiv	
⋮	
netID	
Name	202-n
Indiv	
⋮	
netID	

(b)

MenuID	211
MenuID_OLD	
AbilityID	
AbilityID_OLD	
TermFlag	212-1
ItemFlag	
Status	
TargetNum	
Damage[1]	212-2
⋮	
Damage[k]	
Weapon[1]	
Weapon[2]	212-3
CardFlag[1]	
CardFlag[2]	
LawID[1]	
LawID[2]	212-n
TermFlag	
ItemFlag	
⋮	
LawID[2]	212-n
⋮	
TermFlag	
ItemFlag	
⋮	212-n
LawID[2]	

[FIG. 6]

301

LAW UNIQUE ID	RULE NAME	DETERMINATION TYPE	offset	RED CARD CONDITION	PENALTY TYPE
001	ITEM PROHIBITION	Category	ITEM	STONE_TERM	LOST_TREASURE
002	FIRE PROHIBITION	Ability	FIRE	STONE_TERM	LOST_GIL
003	BLIZZARD PROHIBITION	Ability	ICE	STONE_TERM	LOST_GIL
004	THUNDER PROHIBITION	Ability	THUNDER	STONE_TERM	LOST_GIL
005	SWORD PROHIBITION	Arm	CATEG_SWORD	STONE_TERM	LOST_TREASURE
006	SPEAR PROHIBITION	Arm	CATEG_LANCE	STONE_TERM	LOST_TREASURE
007	BLADE PROHIBITION	Arm	CATEG_BLADE	STONE_TERM	LOST_TREASURE
008	DAMAGE 20 ↑	Damage	20	NONE	STAT_DOWN
009	DAMAGE 50 ↑	Damage	50	NONE	STAT_DOWN
010	DAMAGE 100 ↑	Damage	100	NONE	STAT_DOWN
*	*	*	*	*	*
*	*	*	*	*	*

302

STAGED LAW ID	STAGE 1	STAGE 2	STAGE 3
001	008	009	010
002	001	003	004
003	002	001	007
004	005	006	001
*	*	*	*
*	*	*	*
*	*	*	*

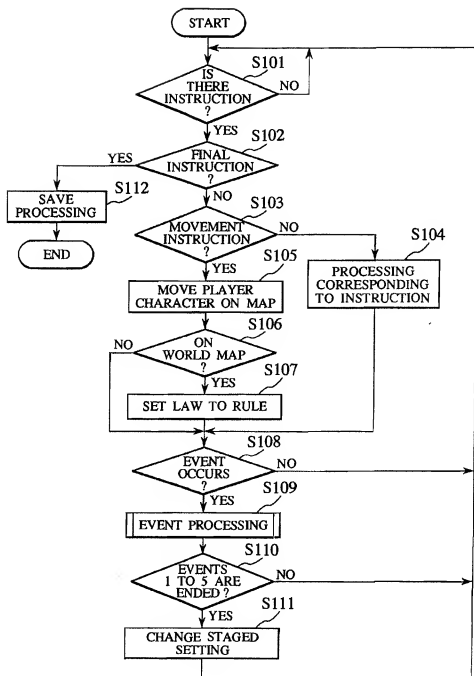
(a)

(b)

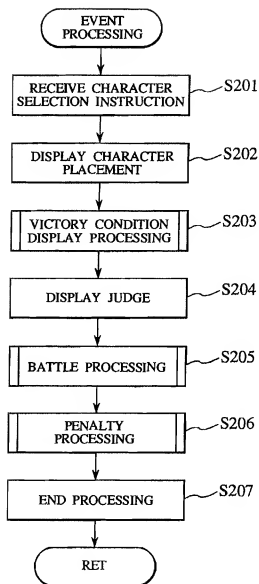
[FIG. 7]

PENALTY TYPE	CONTENTS	APPLYING OBJECT	YELLOW CARD	RED CARD
FINE (LOST_GIL)	GIL IS CONFISCATED (WHEN IT IS NOT PAID, SEND TO PRISON OF THE SAME WEIGHT)	ENTIRETY	100	1000
STATUS DOWN(STAT_DOWN)	AT LEAST ONE OF MAXIMUM HP, MAXIMUM MP, PHYSICAL ATTACK, PHYSICAL DEFENSE, MAGIC ATTACK, MAGIC DEFENSE, AND SPEED IS REDUCED(AMOUNTS CORRESPONDING TO ONE LEVEL ARE UNIFORMLY DECIDED)	PLAYER CHARACTER	ONE TYPE	TWO TYPE
CONSUMPTION ITEM CONFISCATION (LOST_TREASURE)	CONSUMPTION ITEM IS CONFISCATED	ENTIRETY	ONE ITEM IS CONFISCATED FROM LIGHT ITEM GROUP AT RANDOM	ONE ITEM IS CONFISCATED FROM HEAVY ITEM GROUP AT RANDOM

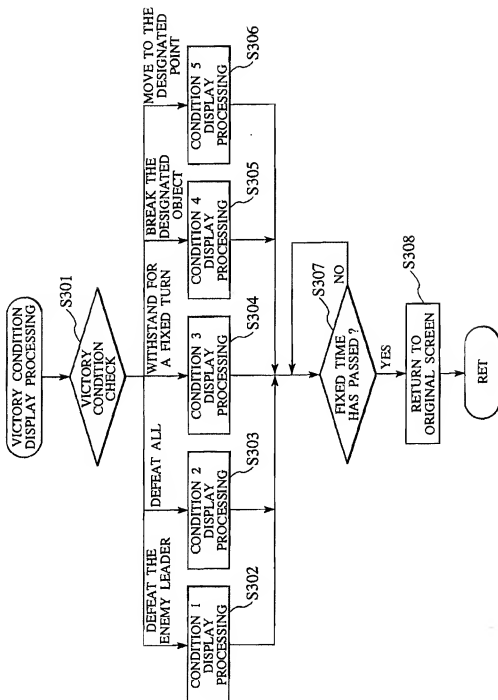
[FIG. 8]



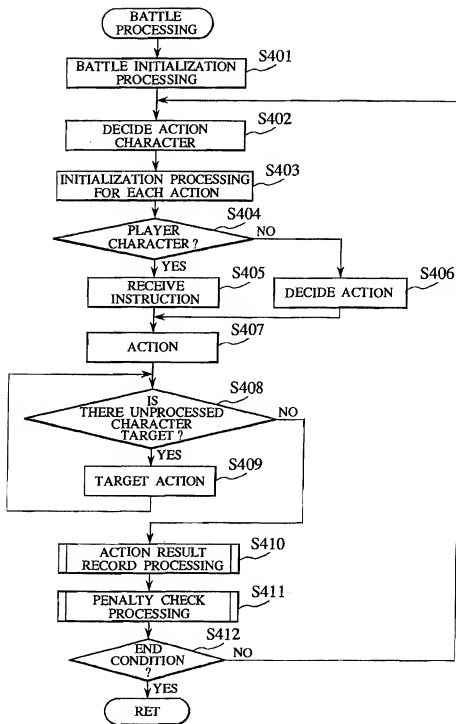
[FIG. 9]



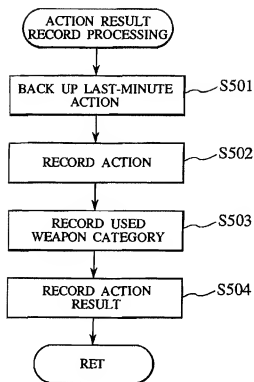
[FIG.10]



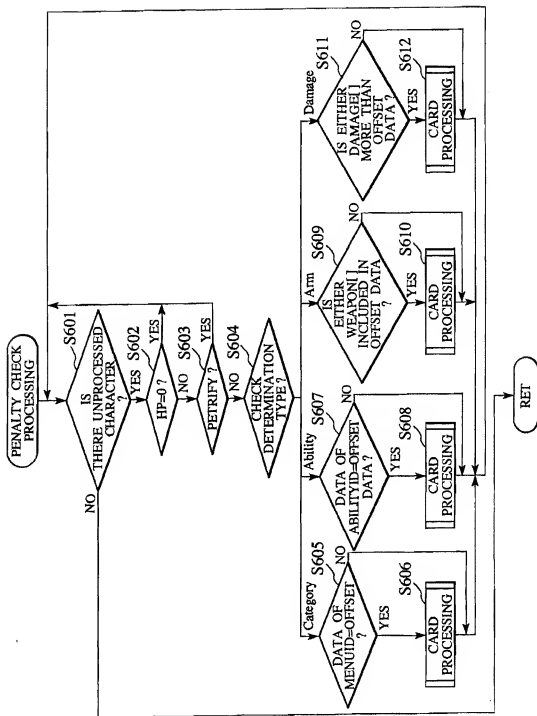
[FIG. 11]



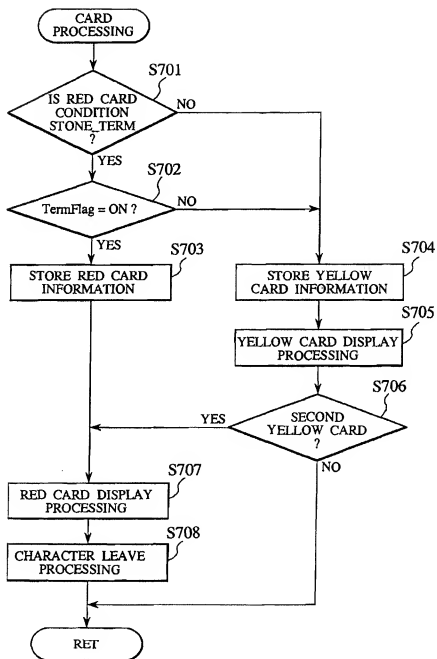
[FIG. 12]



[FIG. 13]



[FIG.14]



[FIG.15]

